

Значимый контекст рассуждений в задаче планирования

Трофимов Игорь Владимирович¹

Автоматическое планирование – задача высокой вычислительной сложности. Универсальные классические планировщики, опирающиеся на эвристики локального поиска, не способны решать эту задачу в предметных областях с большим количеством объектов. В этой работе представлен метод, позволяющий существенно сократить пространство поиска за счет выделения значимого, с точки зрения конкретной задачи, контекста в рамках описания предметной области.

Ключевые слова: планирование, домен, контекст, ассоциации, обучение.

Введение.

В работах [1][2][3], посвященных теории сложности задачи планирования, было показано, что даже при очень сильных ограничениях задача интеллектуального планирования имеет неполиномиальную сложность. В общем случае планирование является EXPSPACE-полной задачей. Свои результаты авторы этих работ продемонстрировали на примерах, в которых использовались описания действий, основанные на STRIPS-формализме [4].

Легко показать, что задача планирования в пространстве состояний относится к классу EXPSPACE. Пусть n – количество атомарных высказываний о мире, которыми оперирует планировщик (предполагается, что все атомарные высказывания независимы и могут быть подвергнуты воздействию, т.е. менять значение истинности). Каждая такая атомарная формула может быть либо истинной, либо ложной в конкретном состоянии модели мира. Таким образом, общее количество возможных состояний модели мира равно 2^n . В худшем случае, выполняя поиск в пространстве состояний, мы можем обойти все эти состояния.

С практической точки зрения интерес представляют задачи с большим значением n . Мир многогранен и содержит огромное количество объектов, их свойств и атрибутов. Компьютерная модель должна отражать все свойства, которые потенциально могут быть значимы для решения определенного класса задач. А если речь заходит о решении широкого спектра задач в рамках одной предметной области, то количество факторов, которые нужно принимать во внимание, становится очень большим.

Очевидно, что при таких (2^n) размерах пространства поиска не имеет смысла пытаться применять полный перебор для решения задачи планирования. Единственным разумным решением является сосредоточение усилий на разработке сложных эвристик, позволяющих в тех или иных предметных областях сокращать пространство поиска до приемлемых размеров.

В этой работе рассматривается метод, который позволяет значительно сузить пространство поиска и, таким образом, сделать решаемыми довольно сложные задачи.

* Работа выполнена при финансовой поддержке РФФИ (проект № 03-01-00853а)

¹ 152020, Ярославская обл., г. Переславль-Залесский, м. Ботик, ИПС РАН, igor@warlock-98.botik.ru

Об эвристиках в планировании

Большинство современных планировщиков используют эвристики, основанные на выборе (в каждом узле дерева поиска) наиболее перспективной ветви поиска. Для этого они запускают процесс планирования от текущего узла дерева поиска вперед на небольшую глубину (локальный поиск) и выбирают дальше ту ветвь, которая в наибольшей степени приближает планировщик к цели. Очевидно, что такой подход чреват попаданием в локальный экстремум. То есть мы можем погнаться за целью путем, который, в конечном счете, приведет в тупик. Вероятность попадания в такую ситуацию зависит от самой задачи и глубины локального поиска. Таким образом, чем более мощными вычислительными ресурсами мы обладаем, тем более глубоким можно сделать локальный поиск, и тем более надежной становится система. В качестве примеров таких планировщиков можно привести современные системы HSP [5] и FF [6].

Еще одним перспективным методом является настройка системы на решение определенного класса задач в данной предметной области (обучение). Изучив ряд готовых решений, система в дальнейшем сама может решать схожие (по структуре решений) задачи. К настоящему моменту выполнено множество попыток применить обучение для настройки планировщика на предметную область, начиная от макрооператоров для STRIPS [7] и заканчивая современными подходами, пользующимися методологией EBL (explanation-based learning) [8] и CBR (case-based reasoning) [9][10]. В этих системах основное внимание уделяется "методам" решения задач. Планировщики обучаются типичным способам решения конкретных проблем в рамках данной предметной области. В дальнейшем эти "методы" (наборы действий) могут быть использованы для решения аналогичных задач, но, например, оперирующих другими объектами.

Метод решения по своей сути является абстракцией, его нужно к чему-то применить. Всякий метод оперирует какими-то объектами предметной области и отношениями между ними. Если объектов и отношений, к которым можно применить метод много, то мы приходим к поиску на множестве допустимых объектов.

Однако не только методы могут быть целью обучения. В рамках заданной предметной области было бы полезно выделять *значимый контекст рассуждений* – набор объектов, которые необходимо принимать во внимание при решении заданной задачи. При этом можно воспользоваться как методом решения, полученным в результате обучения (адекватным), так и любым общим методом (универсальным). Обученная таким образом система довольно жестко привязывается к набору объектов, фигурирующих в предметной области. Но при этом очень сильно сужается пространство поиска. К тому же, такой подход позволяет оперировать специфичными свойствами объектов в данной конкретной предметной области.

Причиной того, что такой вид обучения не получил должного внимания, может оказаться исторически сложившаяся форма разбиения предметной области на домен планирования и задачу планирования. В STRIPS и его современных, более развитых аналогах (например, PDDL [11][12]) доменом планирования считается множество возможных схем действий (хотя в PDDL в описание домена входит гораздо большее число сущностей, мы не будем их здесь рассматривать, т.к. они не имеют отношения к теме). А множество объектов предметной области является элементом задачи. Это дает определенную гибкость в том смысле, что мы одно и то же устройство (представленное набором действий) можем использовать в разных средах (представленных набором объектов и связями между ними). Но при этом мы теряем возможность манипулировать знаниями об объектах в каждой конкретной среде.

О целесообразности применения значимого контекста рассуждений

Рассмотрим примеры, когда выделение и использование значимого контекста рассуждений может оказаться полезным. Существует множество достаточно статичных предметных областей, в которых знания об объектах играют ключевую роль при принятии решений. В частности, примером может послужить транспортная задача.

Пусть у нас есть множество маршрутов транспортных средств. Количество этих маршрутов может быть довольно большим. Кроме того, сами транспортные средства могут быть разнотипны. Будем решать задачи перемещения людей из одного месторасположения в другое.

Мы можем описать действия транспортировки в нотации PDDL, например, следующим образом:

```
(:action fly
:parameters (?from, ?to – location, ?who – person)
:precondition (and (at ?who ?from) (flight ?from ?to) (not (= ?from ?to)))
:effect (and (not (at ?who ?from)) (at ?who ?to)))
```

```
(:action train_transportation
:parameters (?from, ?to – location, ?who – person)
:precondition (and (at ?who ?from) (train_route ?from ?to) (not (= ?from ?to)))
:effect (and (not (at ?who ?from)) (at ?who ?to)))
```

```
(:action taxi_transportation
:parameters (?from, ?to – location, ?who – person)
:precondition (and (at ?who ?from) (not (far_from ?from ?to)) (not (= ?from ?to)))
:effect (and (not (at ?who ?from)) (at ?who ?to)))
```

Этот список может быть продолжен. Здесь первое действие соответствует перелету на самолете, второе – поездке на поезде, третье – не такси. Предикат `at` – отражает местоположение объекта (человека), `flight` и `train_route` – наличие рейса на самолете и поезде соответственно, `far_from` – расположение объектов на большом расстоянии.

Кроме описания действий, в предметной области описывается множество рейсов транспортных средств и множество географических точек (с качественным указанием расстояний), между которыми выполняются рейсы. Еще раз отметим, что эти сущности достаточно статичны. Их все следовало бы описывать в поле `constants` описания домена в PDDL нотации. Но, к сожалению, язык не допускает описания отношений в поле констант. Множество персон в данном случае, динамический аспект. Их разумно описывать как элемент задачи в PDDL нотации.

Теперь рассмотрим задачу переезда персоны `X` из одного города в другой. Пусть `X` находится на железнодорожном вокзале города Ярославль и ему требуется попасть в Нью-йоркский аэропорт им. Кеннеди. Обычно эта задача решается так. Едем на поезде до Москвы, далее на такси в аэропорт Шереметьево и на самолете до Нью-Йорка.

Взглянем на эту задачу глазами планировщика. Если мы идем от цели (`at X, New-York_Kennedy_Airport`), то при данном наборе действий у нас есть множество способов попасть в данный аэропорт. Мы можем приехать туда, например, на такси из близлежащего населенного пункта или прилететь на самолете из Зимбабве. Однако с практической точки зрения при поезде из Ярославля нет смысла рассматривать все эти направления. В 99% случаев нас будет интересовать только авиарейс Москва – Нью-Йорк.

В аналогичную ситуацию мы попадаем, если воспользуемся прогрессией в качестве стратегии поиска планировщика. Из Ярославля можно заехать, например, в деревню Гадюкино, что ни на йоту не приблизит нас к цели.

При большом количестве пересадок эвристический локальный поиск может не спасти. Однако может помочь обучение. Если мы будем учиться только методам, что и реализовано в большинстве систем, использующих CBR, то мы получим набор типичных решений задачи транспортировки без привязки к предметной области. Т.е. мы будем обладать множеством последовательностей действий, которые встречались в обучающей выборке. Но, даже обладая правильным абстрактным планом (а его еще надо найти среди множества прецедентов) вида

`train_transportation(?x, ?y, ?person), fly(?y, ?z, ?person),`

мы можем получить множество нетипичных (как правило, нерелевантных) решений, таких как "поездом до Владивостока, а там в Нью-Йорк".

Этот пример отчетливо демонстрирует неспособность современных методов планирования справляться с задачами такого рода. Таким образом, возникает необходимость искать какой-то новый способ ограничения пространства поиска.

Одним из возможных путей видится введение понятия релевантных сущностей с точки зрения задачи. Сузив контекст рассуждений до релевантного с точки зрения конкретной задачи, мы можем в дальнейшем быстро получить "хорошее" решение при помощи локального поиска или CBR. В примере с поездкой из Ярославля в Нью-Йорк нет смысла рассматривать маршруты на Зимбабве. Зато Москва является релевантным объектом с точки зрения данной задачи.

Технология

Каким образом можно сузить контекст рассуждений? Целесообразным кажется использовать подход, предлагаемый CBR. Только в качестве прецедентов нужно хранить не методы решения задачи, а множество наиболее релевантных объектов и действий с точки зрения данной задачи. Таким образом, по каждой постановке задачи (начальным условиям и цели) мы будем получать некоторое множество объектов (и, возможно, действий) с использованием которых и будет вестись поиск. Для примера с поездкой из Ярославля в Нью-Йорк релевантными будут объекты: Ярославский ж.-д. вокзал, Ярославский вокзал в Москве, аэропорт Шереметьево, Нью-йоркский аэропорт Кеннеди; и действия: поездка на поезде, поездка на такси, полет на самолете (но никак не плавание на яхте). Бесспорно, релевантными будут и все объекты, упомянуты в постановке задачи.

Методология CBR предполагает хранение прецедентов в виде индексированного списка. Индекс отображает постановку задачи в решение. Естественно этот список не хранит решения всех задач. Как поступать, если мы сталкиваемся с новой задачей, которой еще нет в индексе? Обычно в индексе ищется наиболее похожая задача и соответствующее ей решение используется в качестве основы для построения нового решения. Т.к. мы оперируем множествами релевантных сущностей, мы можем взять целый набор похожих задач (превышающих некоторый порог схожести) и считать релевантным контекстом объединение соответствующих множеств (а может даже какую-то выборку из этого объединения). Например, у нас среди прецедентов могут храниться наборы релевантных объектов для задачи поездки из Ярославля в Магадан (через Москву) и поездки из Магадана в Пекин. Если мы теперь захотим попасть из Ярославля в Пекин, то в контексте рассуждений окажутся Ярославль, Москва, Магадан и Пекин. Если нет прямого рейса Москва – Пекин, то будет найдено решение: транзит через Магадан.

Следующий вопрос, который следует рассмотреть, это способ построения библиотеки прецедентов. В этих целях можно воспользоваться методами машинного

обучения. Безусловно, допустимо обучение по обучающей выборке. Кроме того, сама методология CBR предполагает накопление опыта. По мере получения новых решений (прецедентов), можно сохранять их, как отдельные элементы библиотеки прецедентов или модифицировать (например, обобщать или дополнять) уже имеющиеся элементы. Видимо, имеет смысл применить оба подхода: обучающую выборку для начального заполнения библиотеки прецедентов и стандартную методологию CBR для уточнения и поддержания ее в актуальном виде.

После того, как принято решение сохранить полученный план в качестве прецедента, нужно определить, какие сущности должны войти в контекст, соответствующий решенной задаче. Безусловно, туда должны попасть все действия и "статичные" объекты, упомянутые в плане. Нужно ли включать в контекст какие-то еще объекты? Видимо это зависит от предметной области и наших требований к планировщику. Если мы хотим получать существенно новые решения, то возможно имеет смысл пополнить этот контекст еще какими-то объектами. Принятие решения по этому вопросу можно, например, возложить на эвристики, сети ассоциаций или другие методы.

Интересно было бы также исследовать, можно ли применять выявление значимого контекста в каждом узле дерева поиска. Попадая в новый узел (состояние), мы фактически получаем новые начальные условия для задачи (если мы ведем поиск от начального состояния к цели). Возможно, выделяя значимый контекст уже для этой новой задачи, мы сможем находить более эффективные решения.

Выводы

Планирование, опирающееся на выявление значимого контекста рассуждений, может дать существенный выигрыш в скорости поиска и качестве построенных планов в предметных областях со статичной структурой. Более того, иногда такой подход будет даже единственным возможным решением. Выделение значимого контекста колоссально сужает пространство поиска. Ведь даже удалив один объект из рассмотрения, мы уменьшим n на величину, соответствующую количеству упоминаний этого объекта в атомарных формулах, описывающих предметную область. Возможно, в ряде предметных областей этот метод сможет даже стать ключом для перехода на прикладной уровень.

Список литературы

1. T. Bylander. Complexity Results for Planning. IJCAI'91, стр. 274-279.
2. T. Bylander. The Computational Complexity of Propositional STRIPS Planning. Artificial Intelligence 69(1-2), 1994, стр. 165-204.
3. K. Erol, D. S. Nau, V. S. Subrahmanian. Complexity, decidability and undecidability results for domain-independent planning: A detailed analysis. Tech. Rep. CS-TR-2797, UMIACS-TR-91-154, SRC-TR-91-96, Computer Science Department and Institute for Systems Research, University of Maryland, 1991.
4. Fikes R.E., Nilsson N.J. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. IJCAI'71, стр. 608-620.
5. B. Bonet, H. Geffner. HSP: Heuristic Search Planner. Entry at the AIPS-98 Planning Competition, Pittsburgh 6/98.
6. J. Hoffmann, B. Nebel. The FF Planning System: Fast Plan Generation Through Heuristic Search. Journal of Artificial Intelligence Research, том 14, 2001, стр. 253 - 302.
7. Fikes R.E., Hart P.E., Nilsson N.J. Learning and Executing Generalized Robot Plans. Artificial Intelligence, том 3, 1972, стр. 251-288.

8. Minton, S. Learning Effective Search Control Knowledge: An Explanation-Based Approach. Kluwer Academic Publishers, Boston, MA.
9. Rousu J. Case-based Planning. In Seminar on Knowledge Engineering, Helsinki University of Technology, 1997.
10. Spalazzi L. A Survey on Case-Based Planning. Artificial Intelligence Review, том 16, 2001, стр. 3-36.
11. Ghallab M., Howe A., Knobloch C., McDermott D., Ram A., Veloso M., Weld D., Wilkins D. PDDL - The Planning Domain Definition Language (version 1.2). AIPS'98, 1998.
12. Fox M., Long D. PDDL 2.1: An Extension to PDDL for Expressing Temporal Planning Domains. Journal of Artificial Intelligence Research, том 20, стр. 61-124, 2003.